



Software Product Development: Transforming Art to Science



The functional capabilities of today's aircraft and weapon systems are increasingly dependent on the software resident in these systems. Expanding the functionality to meet the demands of an ever-changing environment drives a seemingly insatiable demand for software development resources. The realities of our fiscal and personnel environments preclude a one-for-one approach to meeting those demands. Like many other organizations, the Naval Air Systems Command (NAVAIR) has embarked on a journey to transform the way it acquires and develops software. Nothing less than a transformation will yield the efficiencies we need to support our customers: the sailors and Marines who utilize our products to protect and defend our nation.

The military, civilian, and contractor teams supporting NAVAIR have produced software products that have met user expectations for many years. Their dedication and hard work has allowed successful deployment of very complex weapon systems. However, I would have to characterize the efforts as more art than science. Many of our most significant releases have been based on the heroic actions of team members. While laudable, such an approach is not sustainable in a steady-state environment, much less in one of significantly increasing demands.

Our quest for transformation is based on some important changes. Two of them are structural in nature. The other, tied closely with the structural changes, is cultural. The initial structural change – already under way – groups our existing 52 standalone product teams into four Mission Area Teams (MATs), each with a single leader responsible for working with many customers and sponsors to deliver products efficiently by utilizing shared resources. This specific change is driving the largest cultural change. Prior to this transformation, each separate product team leader carried full resource and product responsibility. This arrangement did little to enhance efficiency across the teams as resource sharing could lead to perceived increases in risk to the separate product teams.

The second structural change is more germane to CROSSTALK. NAVAIR has decided to embrace the Software Engineering Institute's (SEISM) Capability Maturity Model[®] Integration (CMMI[®]) model as overall architecture to guide process improvement within the MATs. NAVAIR has a long history of utilizing SEI models, and has achieved significant success in process improvement using the CMM[®] for Software (SW-CMM), but it has been a fragmented approach dependent on the process improvement philosophy of each individual team. Within CMMI, processes are developed at the organizational level with tailoring guides to allow individual project teams within the organization to utilize the processes. We have established the MATs as the CMMI organizational focus, and are working to document the organizational level processes and tailoring guides for each applicable process area. This single structural change will allow much more flexibility for resource sharing between teams as each MAT member will utilize the same processes regardless of his or her specific project team. This overarching process improvement culture should bring steadily increasing rigor to our software development practices, allowing us to accurately predict and execute resource requirements and project risks, precluding the need for continuous heroic efforts.

Each NAVAIR product team selected combinations of tools that met its specific needs to do software process improvement, including SW-CMM, CMMI, Earned Value Management System, High Performance Organization training, and Team Software ProcessSM (TSPSM). Since the theme of this edition of CROSSTALK is Personal Software ProcessSM/TSP, I would like to reference some specific examples of how TSP has significantly helped accelerate organizational software process improvement in NAVAIR. SEI data shows an average of six years to achieve SW-CMM Level 4. At NAVAIR, we proudly point to three organizations at multiple locations that successfully used TSP to achieve SW-CMM Level 4 in less than three years: AV-8B (2003), P-3C (2004), and E-2C (2005)!

The articles within this issue are intended to give you a flavor of the successful use of TSP to further software process improvement efforts. I hope you enjoy learning from your colleagues' past efforts and future plans.

Terrence Clark
Director, Software Engineering
Naval Air Systems Command